스마트카드의 기준에 관한 연구

대전대학교 정보통신공학과
이 승혁*　황 선태**

* 대전대학교 정보통신공학과 석사과정

** 대전대학교 정보통신시스템공학부 교수

H/W 구 조 및 특 징

- 1. Die Size : 25 mm² 이내.
- 2. Feature Size : Sub-micron Technology(1-um 미만).
- 3. Vcc : 3V, 5V.
- 4. Contact(최소 2x1.7mm²) : CLK, RST, Vcc, GND, I/O.
- 5. CPU : 8-bit, ..., 32-bit RISC Processor.
- 6. ROM : 16 - 32 Kbytes; COS나 불변의 Data 저장.
- 7. RAM : 1 Kbytes 미만; CPU의 Scratch Memory.
- 8. EEPROM : 8 - 16 Kbytes; Card Id #, PIN, Balance, Credit Limit 등 저장.
- 9. Flash Memory 이용.

ISO 7816 규정

- 1. 11개의 Parts로 구성:
- Part 3 - IFD와 Card간의 전송 Protocol 지정.
- Part 4 - IFD와 Card간의 기본 명령어 지정.
- 2. Class A의 Vcc : 5 V, Class B의 Vcc : 3 V.
- 3. Card 동작 속도 : 1-5 MHz at Class A, 1-4 MHz at
- Class B.
- 4. Clock Duty Cycle : 40 - 60 %.
- 5. 단말기-Card 간의 초기화: Reset/ATR/PTS 사용.

ISO 7816 규정 (계속)

- 6. APDU(Application Protocol Data Unit) :
  IFD-Card 간의 전송 Message.
  명령어를 포함한 4-byte Header + Lc, Data, Le를
  포함하는 Body.

- * Lc = 명령어의 Data Byte 수;
- Le = 응답 Message의 예상되는 최대 Byte 수.

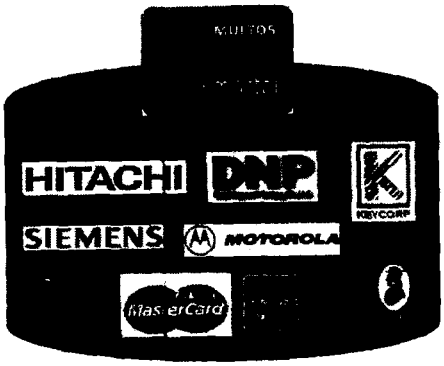
스마트카드 표준화 동향

- ISO 7816: Identification Cards - Integrated Circuit cards with contacts
- ISO 8583: Financial transaction cards -- Messages between the integrated circuit card and the card accepting device
- ISO 10202: Financial transaction cards – Security architecture of financial transaction systems using integrated circuit cards
- ISO 10536: Identification cards -- Contactless integrated circuit(s) cards
- ISO 14443

스마트카드 표준화 동향 (계속)

- EMV'96
- C-SET
- PC/SC
- MULTOS
- JAVA

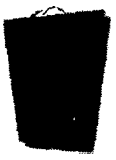
JAVA & MULTOS



JAVA & MULTOS (계속)

공통점

- Multi OS 지원
- 다양한 Application 탑재 가능
- 컨소시움 형태로 추진
- OS와는 별도로 Application 개발 가능
- SPEC 공개

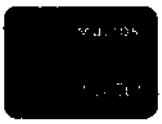


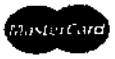
JAVA & MULTOS (계속)

차이점

- Master Card 가 주도
- 일반 C 언어로 구현
- MULTOS 운영체제 에서 실행됨
- JAVA CARD 인터페 이스 지원
- MONDEX 전자지갑

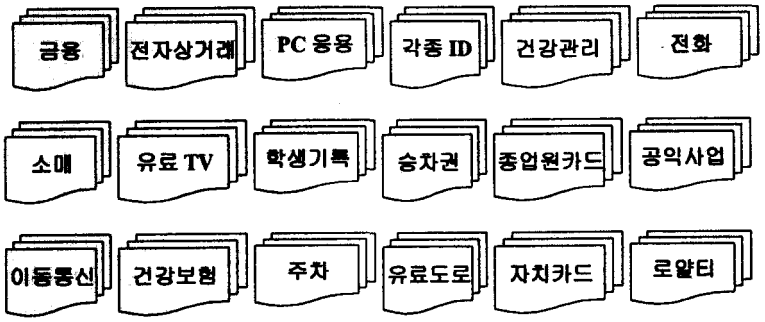
- VISA 가 주도
- JAVA 언어로 구현
- 개별 운영체제에서 실행됨
- 자동메모리 관리 자동 가비지 컬렉션 코딩시간 크게 단축
- Cyberfelx 카드





VISA

스마트카드의 활용



이동통신 분야(GSM)

- 1. GSM(Global System for Mobile Communications) : TDMA 방식의 Digital 이동통신 유럽 표준.
- 2. IFD에 스마트카드 형태인 SIM(Subscriber Identity Module)을 삽입 후사용 - 통화 내용의 암/복호화 및 사용자 인증.

이동통신 분야(GSM) (계속)

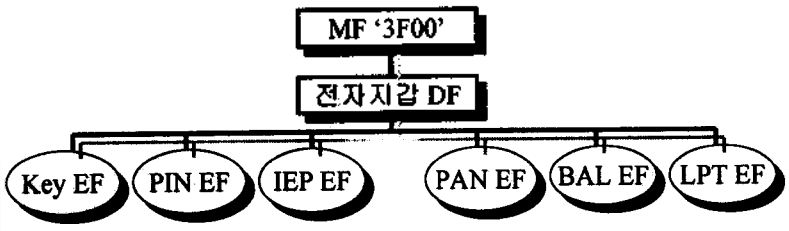
- 3. 4-GSM D/Bs 사용.
- 4. GSM SIM에서의 보안 : 3가지 보안 Algorithm사용
  a. A3 Algorithm - 망에 대한 SIM의 인증을 위한 용도.
  b. A5 Algorithm - Base 기지와 Mobile 기지간 정보암호화.
  c. A8 Algorithm - A5에서 사용되는 암호키를 생성.

전자지갑 분야

- 1. 전자지갑 : 단말기를 통한 가치 저장, 구매 지불, 이체, 환불 등이 가능한 스마트카드.
- 2. 암호화 되어 저장되는 정보 : 저장 가치, 발행 은행, 사용자 사항, 지불 은행 등.

전자지갑 분야 (계속)

- 3. File 구조 : 한 개의 MF와 하나의 전자지갑 DF 밑에 다수의 EF(Data Files)가 링크 되어 있다.



COS 설계 기준

- COS의 기능 :
- 1. 응용 프로그램의 H/W 접근 관리.
- 2. Smart Card의 기능 결정 -
- a) 카드와 단말기 간의 데이터 송수신
- b) 명령어 수행 제어
- c) 데이터 관리
- d) 암호 알고리듬 수행

COS의 명령어 처리 과정

- 1. Serial I/O Interface - 명령어 전송
- 2. I/O Manager - Error Check
- 3. Secure Messaging Manager - 명령어 해독
- 4. Instruction Interpreter - 명령어 해석
- 5. Return Code Manager - 문제시 Return-code 생성하여 I/O Manager 거쳐 단말기로 송신
- 6. Logical Channel Manager - 명령어를 위한 Channel 선택(Manage Channel)
- 7. Status Automaton - 명령어 Status Check

COS의 명령어 처리 과정(계속)

- 8. Code Interpreter - 명령어내의 Code를 수행함
- 9. File Manager - Data 주소 관리/
  Access 조건 Check
- 10. Memory Manager - EEPROM 관리

COS Design Principles

- 1. Modular Design - 1Kbytes 미만의 Module로
  설계/Coding/Test. Module 간 상호 독립적.
- 2. Core 부분은 Assembly어 사용, File Manager
  나 State Automaton Module 등은 C나 Java로
  개발 - 개발 시간 단축, Test 용이. 추가의
  ROM/RAM 필요.
- 3. EEPROM의 Life-cycle - 십만회 정도
  Write/삭제. *Atomic Routine*의 경우 Buffer
  위치 변동.

COS Design Principles (계속)

- 4. Memory 사용 규칙:
  - a) EEPROM이 ROM보다 1-bit당 4배의 면적 필요 - Cost와 면적 문제.
  - b) EEPROM은 Access Time이 길다 - 1 ms/1 byte 이상.
  - c) RAM의 I/O Buffer(보통 100 bytes)보다 큰 Data 핸들링 - RAM의 모든 Data를 EEPROM에 Copy후 RAM 전체 사용.
    --- RAM의 용량 증가 필요.

COS Design Principles (계속)

- 5. File 특성 - Memory 사용 최소화.
  - Header는 File 구조/Access 조건 포함.
  - Body는 Data 포함.
  - Header와 Body는 서로 다른 위치.
  - Header는 사용자가 변동 금지.
  - 고유의 File ID를 이용해 Access.
  - 한 개의 응용 Program당 한 개의 DF.

COS Design Principles (계속)

- 6. File 구조 :

  a) Transparent - Offset을 이용해 Byte나 Block 단위로 Access.

  Block이나 Offset크기는 Memory 크기에 좌우됨.

  Digitized Picture 저장에 사용.

COS Design Principles (계속)

- b) Linear Fixed/Variable - 254개 이하의 Record 집합.

  254 byte 이내의 정해진 길이(Fixed),

  254 byte 이내의 임의의 길이(Var.).

  Variable인 경우 Length Field 필요.

  Variable 방식은 제한된 Memory 사용을 극대화.

COS Design Principles (계속)

- c) Cyclic - 254개 이하의 Record 집합.
  - 254 byte 이내의 정해진 길이 Record.
  - 현재 Record를 기준으로 이전/다음 Record로 Access.
  - 가장 오랜 Record 제일 먼저 교환.
  - 작은 Memory의 효율적 이용.
  - Protocol File에 이용.

COS Design Principles (계속)

- 7. Atomic Routine 개념 - 비정상적인 작업 종료 시 S/W적인 해결책.
  - State Flag과 Buffer가 EEPROM에 필요.
  - Buffer에 원래 Data의 주소와 내용을 Copy.
  - 문제시 Buffer 내용을 COS가 Restore.
  - *문제점 : EEPROM의 수명 단축.*
  - *처리 시간 지연(2, 3배).*
  - *해결책?*

COS Design Principles (계속)

- 8. 보호 기능 - IC Chip은 VHDL로 Design.
  에폭시 수지의 물리적인 제거 시 칩 손상.
  전기적인 시도 시 EEPROM 내용 등은
  특수 논리회로에 의해 자폭기능
  (Kill-bit-logic).

COS Design Principles (계속)

- 9. Flash Memory - ROM 대체 가능.
  ROM Masking 시간을 제거.
  C 등으로 용이하게 Program 가능.
  Download Disable/Enable 가능.
  10년간 1,000번 이상 Write 가능.
  Access Time 빠름 : 1 ms/1 byte 이내.

COS의 Test

- 1. H/W Test - 기능 Test와 전기적인 특성 Test.
  - ATPG를 이용한 Test Pattern 생성[Netlist].
  - Test Coverage Evaluation.
  - BIST : Test 회로 내장. Area Overhead 고려.
  - DFT : Test 용이 하도록 회로 Design.
  - CPU : 기능 Test.
  - Memory : Bit의 Read/Write/Clear Check.
    - 이웃 Cell과의 접속 관계 Test.
  - IDDQ Test : 전기적 특성 Test.
  - Card's Body Test : ISO/IEC 10373에 정의.

COS의 Test (계속)

- 2. S/W Test - ROM에 저장된 Program의 기능을 Release 이전에 철저히 확인.
  - 미국 : 1981년 NCSC(US National Computer Security Center) 설립.
    - 1985년 Orange Book 출간(Trusted Computer System Evaluation Criteria)
  - --- TCSEC가 범 세계적인 Model.
  - 유럽 : 1990년 ITSEC(Information Technique System Evaluation Criteria) 발표.
  - *무자격자의 Data나 기능에 대한 Access 방지.*

COS의 Test (계속)

- *유럽 표준 - EN 1292 (ATR과 T=1 Protocol의 일부 Test 정의).*

  COS Test : Data 전송 Test부터 시작.

  a) Data Transmission Test ATR, PTS 등

  b) Instruction Test

  c) File Test MF, DF, File 특성 등

  d) Routine Test Micro/Macro Instruction 순서

결 론

- 1. 스마트카드 응용의 성패는 키 관리를 포함한

  COS 및 Protocol 보안 문제.
- 2. 다 기능 스마트카드 구현을 위한 OS의 개발.
- 3. 다 기능 COS를 위한 고집적의 새로운 메모리 개발.
- 4. 빠르고 완벽한 H/W 및 S/W Test 기법/Tool 개발.